ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/337038964

Practical Key Recovery Model for Self-Sovereign Identity Based Digital Wallets

Conference Paper · August 2019

DOI: 10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00066

CITATIONS
0

READS
100

3 authors:

Reza Soltani
York University
5 PUBLICATIONS **24** CITATIONS
SEE PROFILE

Uyen Trang Nguyen
York University
71 PUBLICATIONS **914** CITATIONS
SEE PROFILE

Aijun An
York University
197 PUBLICATIONS **3,416** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Internet of Things (IoT) Networks View project

Distributed Deep Learning for Video Processing View project

All content following this page was uploaded by Uyen Trang Nguyen on 24 July 2020.

The user has requested enhancement of the downloaded file.

Practical Key Recovery Model for Self-Sovereign Digital Wallets

Reza Soltani
Lassonde School of Engineering
York University
Toronto, Canada
Email: rts@cse.yorku.ca

Uyen Tran Nguyen, Aijun An
Lassonde School of Engineering
York University
Toronto, Canada
Email: utn@cse.yorku.ca

***Abstract*—Recent years have seen an increased interest in digital wallets for a multitude of use cases including online banking, cryptocurrency, and digital identity management. Digital wallets play a pivotal role in the secure management of cryptographic keys and credentials, and for providing certain identity management services. In this paper, we examine a proof-of-concept digital wallet in the context of Self-Sovereign Identity and provide a practical decentralized key recovery solution using Shamir's secret sharing scheme and Hyperledger Indy distributed ledger technology.**

***Index Terms*—Self-Sovereign Identity, Identity Management, Key Recovery, Blockchain, Digital Wallet, Shamir Secret Sharing**

I. Introduction

Existing identity management models with heavy reliance on centralized repositories of identity data have led to ever increasing security breaches, loss of data and significant cost for all stakeholders, especially identity owners. Moreover, a number of major challenges have led the industry and the academia to explore innovative approaches to the management of digital identity information and cryptographic secrets.

II. Challenges

Major challenges driving the industry and academia are user data ownership, password-based authentication, data fragmentation, client on-boarding and identity breaches.

1) Data Ownership: With over 1.1 billion people lacking official identity [1] and 3.5 people worldwide who are underbanked [2], access to accurate, privacy preserving and secure identity information is essential. Moreover, identity information and secrets such as cryptographic keys and passwords are not always in complete control of the rightful owners. Lack of adequate privacy controls, and visibility to how user identity data is generated, managed and shared by third parties is a major concern. These issues have recently become more visible partly due to new regulations such as General Data Protection Regulation (GDPR).

2) Password-based Authentication: Users continue to use weak passwords. This is partly due to the overwhelming number of passwords that users must manage. An average business user must manage close to 191 passwords [3]. Initiatives such as the use of multi-factor authentication instead of a single factor, or public key cryptography are among the potential solutions in replacing and augmenting passwords.

3) Fragmented Identity Data: Users identity data is fragmented among multiple data repositories. These repositories include governments, banks and health care facilities. Consequently users must establish and manage large number of accounts to interact with these repositories. Moreover, each data repository may potentially follow a different security regimen and privacy policy for protecting their users' identity data.

4) Client Onboarding and Know-Your-Customer: Identity management is an integral part of Know-Your-Customer (KYC), Anti-Money Laundering and Customer Due Diligence processes. Financial institutes such as banks and credit unions rely on credible trustworthy identity information to be able to enhance their recognition of customers identity and credit worthiness and to better manage their liabilities and risks. On the other hand financial customers expect secure, privacy preserving and efficient approaches in interacting with online financial services.

5) Breaches and Identity Fraud: With identity fraud and breaches on the rise, exploration of novel identity management models in which users play a more vivid role is critical.

An identity and access management system (IAM) is a collection of tools, processes and policies used to manage individual identities, their authentication, authorization, roles and privileges, within an organization or across boundaries [4]. The Self-Sovereign Identity (SSI) architecture is a recent iteration of IAM models. The SSI architecture is based on a peer-to-peer model in which the identity owners are the sovereign owners of their identity information, and have control over how they store and manage their identity data. Christopher Allen has provided the ten principles of the SSI model [5]. The SSI model provides a more user-centric approach to identity management in which users are in possession of one or more digital wallets to manage their secret keys and identity information.

The SSI model introduces a number of unique challenges. The adoption of the SSI model depends on addressing these challenges. A notable challenge is related to designing secure, practical and privacy preserving approaches to key management. While traditional identity management models provide

a key management protocol that rely on a trusted third party, in the context of the SSI model the responsibility of key management is assigned to the identity owners themselves. Lack of proper key management protocols in place leads to vulnerabilities, loss of data and fraud.

A. Motivation and Contribution

A digital wallet provides the capability to manage secret keys, identity claims and various other personal information. Practical and privacy preserving key management protocols are integral to the adoption of digital wallets. This is particularly paramount in the context of self-sovereign identity and cryptocurrency. This paper provides a practical key backup and recovery protocol based on threshold secret sharing cryptography. The protocol relies on multiple third party key escrow providers in which each participant holds a share of the user's secret key. The secret key can only be correctly recovered when enough number of participants are involved. The execution of this protocol along with the protocol parameters such as the minimum number of participants required, and the role of the user are entirely configurable by the proposed digital wallet application.

B. Paper Structure

In Section III we present related work around digital wallets, key management and secure multi-party computation. Section IV describes our proposed key recovery model, and Section V provides a high-level description of our digital wallet implementation. Finally, Section VI states the future work and concluding remarks.

III. Related Work

In this section we provide an overview of the building blocks around digital wallets and decentralized key management systems, and a detailed description of common key recovery mechanisms.

A. Digital Wallets

Physical wallets have been around for a long period of time and have evolved since. A wallet holds various items including financial cards, ID cards, cash, receipts, business cards and other items such as pictures and coupons. A digital wallet perform almost all of the functions of a traditional wallet but in a digital context. A digital wallet is system typically running on a smart phone that serves as an electronic version of a physical wallet. A digital wallet is capable of storing various information such as payment information, coupons and in some cases identity credentials. In the context of SSI model, a digital wallet contains various identity information, key pairs, credentials, tokens, and optionally service endpoints. A digital wallet is also able to establish communication with other entities to authenticate or exchange information.

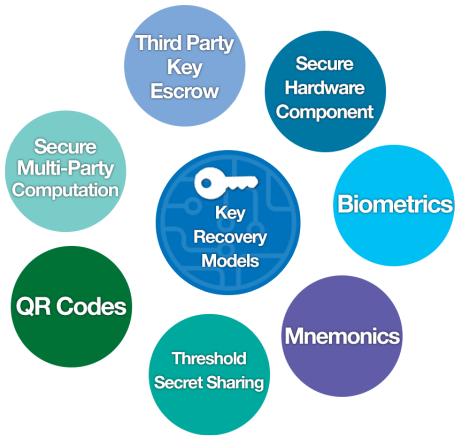

Fig. 1. Common key backup and recovery methods

B. Decentralized Key Management System

Decentralized Key Management (DKMS) is an emerging open standard for managing cryptographic keys in a decentralized architecture so that no single party can compromise the integrity and security of the system as a whole [6]. A DKMS wallet is intended to work with decentralized systems such as DLT and blockchain without a central authority.

C. Common Key backup and recovery methods

Storage and recovery of cryptographic keys is an integral component of decentralized key management systems. As shown in Figure 1 various approaches have been proposed and discussed in the academia and the industry. This section describes a selection of common approaches in more details.

1) Secure Elements and Trusted Execution Environment: Secure Elements (SE) is a combination of software, hardware and protocols within devices to enable secure storage of secret information. Trusted Execution environments (TEE) is a secure and protected processing environment isolated from the rest of processing environment [7]. A TEE component consists of processing, storage and temporal memory capabilities. Using TEE it is possible to store cryptographic keys and performs cryptographic operations, even in the face of situation where the operating system is compromised. Popular TEE implementations include ARM TrustZone and Intel SGX. Finally, hardware security modules (HSM) are isolated hardware components which provide a secure area for storage of private keys and performing cryptographic operations [8].

2) Biometrics: First proposed by Bodo [9], key management based on biometric traits have been studied extensively in the research community [10]. Biometric traits are portable and to a good degree unique and consequently can be used as the seed values in generating cryptographic keys.

3) Secure Cloud Storage and Key Escrows: Storing secret keys with trusted third party repositories (also known as key escrow providers) is another common approach to key backup and recovery. Secure cloud based storage services can be provided by trusted third parties such as financial

institutes, telecommunication providers, government entities and the private sector.

4) Mnemonics and QR Code Technology: To assist with memorization of secret keys, mnemonic words can be derived from secret keys. Mnemonic words are sometimes known as seed phrases. It is also possible to generate paper or digital QR codes of the secret keys for convenience and portability.

5) Secure Multi-Party Computation and Threshold Secret Sharing: Privacy is defined as limitation of the information exposed by the distributed computation to be the information that can be learned from the designated output of the computation [11]. Secure multi-party computation (SMC) [12] provides a privacy preserving method for a group of entities to jointly compute a function over their input while not having the need to share their input with other parties, or with a trusted third party. SMC is a well studied problem in cryptography. Assuming there are three parties Alice, Charlie and Bob with secret inputs x, y and z representing their respective salaries. In order to to find out the highest of the three salaries without revealing to each other their salary they engage in a MPC protocol. The output of the protocol is simply the highest value of the salaries which can then be used by each party to compare against. Mathematically, the problem of finding the maximum value simply translates to $F(x, y, z) = max(x, y, z)$.

In 1986 Yao proposed the two party computation protocol using cryptography [12]. In this protocol two parties can generate a random number $R = PcQ$ such that prime numbers P and Q can only be recovered through joint effort of the two parties. Goldreich et al. [13] further extended Yao's idea by providing a polynomial time algorithm to solve the multi-party (mental game) problem under the assumption of majority of parties being honest. Various applications of SMC have since been introduced. Namely the private information retrieval problem (PIR) [14]. PIR describes a server-client architecture in which the client retrieves the i^{th} bit of a binary sequence from the server without its awareness. On the other hand the server does not reveal the entire bit sequence to the client. Privacy preserving data mining is another application of SMC. Paper [15] introduces an approach in which two parties each with a separate database, intend to jointly perform data mining operations on the union of their data sets without disclosing their databases to one another or any third party.

Nair et al. [16] propose a secure information storage and retrieval model based on SMC. This model relies on Shamir secret sharing [17] to shard sensitive data among multiple databases. The reconstruction of data is performed by the computation agent on the client side.

Clifton et al. [18] describes a toolkit for various privacy preserving SMC based data mining operations including secure sum computation using a secret random number. However in this protocol two adjacent parties party can collude maliciously to compute the random number. To avoid the risk of two parties colluding to determine the secret random number. Sheikh et al. [19] propose a privacy preserving k-secure sum protocol to allow the parties compute the sum while keeping their individual data secret. This is achieved through segmentation of each individual party and generation of random numbers within each segment.

Pathak et al. [20] extends on this work to propose a secure and privacy preserving data mining protocol. In this proposal each party breaks its data block into k number of segments and distributes $k-1$ segments with the other parties. Through a series of interchanges between the parties the total sum of the data is calculated while ensuring the privacy of individual data.

Pinkas [11] provides a detailed discussion of SMC and various privacy preserving data mining techniques.

Early examples of multi-party sharing for RSA signatures is proposed by Santis et al. [21]. Secret threshold sharing and secret segmentation refer to the distribution of secrets into multiple shares. These shares are then disclosed with multiple participants. To recover the secret a pre-determined number of shares must be retrieved from the participants. In other words given a finite set P of participants and a set Γ of subsets of P, the secret threshold scheme defines a threshold t where the secret access model is $\Gamma = \{A \subseteq P : |A| \geq t\}$. In these schemes less than t participants cannot recover the original secret.

In 1979 Shamir [17] and Blakely [22] proposed two variations of threshold based secret sharing schemes. The scheme proposed by Shamir relies on standard Lagrange polynomial interpolation, while Blakley's proposal, also known as vector scheme, relies on the geometric concept of intersecting hyperplanes. An evaluation of Shamir's threshold secret sharing scheme against Blakley's approach is discussed in [23].

Shamir's secret sharing algorithm is described as follows: User V splits a secret S into P shares, such that any combination of $< t$ shares cannot learn the secret S , while any combination of $\geq t$ shares can learn the secret S. To this end the user V constructs a polynomial f of degree $t-1$ such that $f(0) = S$. For example if at least 4 shares are required to reconstruct S then the polynomial will be $f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$. Next, user V computes $f(x_i)$ for all shares $i \subseteq P$ and the resulting pairs $\{x_i, f(x_i)\}$ are distributed to each participant i. In order to reconstruct the secret S the polynomial $f(x)$ should be recreated from the set of pairs provided by the participants. This process can achieved by relying on Lagrange Polynomial Interpolation. The major steps of Lagrange Interpolation are as follows: Given a polynomial $f(x)$ over of degree at most l, and given C such that$|C| = l + 1$, then

$$f(x) = \sum_{i \in C} f(i)\delta(X)$$

where $\delta_i(X)$ is a degree l polynomial such that:

$$\delta_i(j) = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$$

Which can also be represented as:

$$\delta_i(j) = \prod_{j \in C, j \neq i} \frac{X - j}{i - j}$$

Sun et al. [24] propose a key recovery model in which the user's cryptographic key is divided into several fragments, and each fragment is encrypted using a private key derived from the master key. Then each fragment is sent to a different key escrow. To recover the keys, the fragments are retrieved from the key escrow providers and decrypted using the master key.

IV. Proposed Model

In this section we present our decentralized key backup and recovery model based on Shamir's threshold secret sharing algorithm. In our model, user is in possession of a digital wallet capable of producing cryptographic secret keys and performing backup and restoration of the secret keys.

A. Architecture

We model a digital wallet capable of storing sensitive information such as cryptographic keys and identity credentials within the secure element of the mobile device. The digital wallet is able to perform backup and recovery of secret keys in a distributed and privacy preserving approach. By relying on threshold secret sharing algorithm the digital wallet is capable of splitting user's cryptographic keys into multiple shares and sending them to multiple third-parties that are carefully chosen by the user. In this paper these parties are known as key escrow providers. The digital wallet is considered the only entity within our model with access to the entire secret keys. The key escrow providers run a specific digital wallet implementation capable of engaging in the key exchange protocol with users' digital wallets. The key escrow feature can be a service provided by telecommunication providers, banks, or credit unions. Combining multiple key escrow providers create a more neutral and trustworthy system. Relying on providers with opposing interests to further increase the trustworthiness of the system is also an idea that we have entertained. Our architecture supports three operations, namely key generation and registration, key backup and key recovery.

B. Secret Key Generation and Registration

The first step of key management describes the method by which a user generates a cryptographic key pair. The cipher suite used to generate the keys depends on the hardware capabilities of the mobile device, the capabilities of the digital wallet, user's preference and the purpose of the keys. The digital wallet is capable of storing the generated secret keys within the mobile device's secure element. During the registration phase the digital wallet performs a scan of the network to seek available key escrow providers, and allow the user to register them as possible options for the key backup and recovery procedures.

C. Secret Key Backup Process

Upon a successful generation of secret keys, the digital wallet provides the user the ability to create backups of any of the cryptographic keys. Given a cryptographic key S and set of P parties $p_1 ... p_n$, the digital wallet divides the key into a set of shares $\{s_1 ... s_n\}$ for n participants. The user defines the value t as the minimum number of participants required to restructure the secret. This value is chosen based on the user's preference and the availability of key escrow providers.

To support cases where multiple secret keys $S_1 ... S_n$ are present in the wallet, the wallet will sign each share with a temporal key. In other words the wallet will generate a temporal key pair for each secret key and use a specific temporal private key to sign every share s_i for each secret S_j. In the next step the wallet creates a data bundle to be sent to each key escrow provider. For every provider i, the bundle consists of the key share s_i, the total number of key escrow providers n and the threshold value t, in addition to the digital signature generated based on the aforementioned items, and finally the corresponding public key. Once all bundles are prepared, the private key used to generate the signatures is purged. Next, the wallet securely shares each bundle b_i with key escrow provider i. Bundles are encrypted as $E(b_i, k_i)$ where k_i is the public key of the key escrow provider i.

D. Secret Key Recovery Process

To recover the secret key S_i, the digital wallet attempts to establish a secure communication channel with each key escrow provider within set P such that $|P| \geq t$, to obtain the corresponding share in their possession. Once every bundle is retrieved, the wallet performs an evaluation on the bundle to ensure that a) all public keys are same and b) all signatures are valid using the provided public key. Next, the wallet utilizes the shares from each bundle along with Lagrange interpolation to reconstruct the polynomial function $f(x)$. Finally, by evaluating for $f(0)$ the wallet is able to recover the original secret key S_i. The secret is then stored within the mobile device's secure element .

E. Assumptions and Discussion

In the proposed protocol a secure and encrypted communication channel among the digital wallets is assumed. Moreover, we assume the digital wallet is able to successfully authenticate the user, and is the only trusted entity with the ability to generate, backup, recover and store the entire cryptographic keys. Our assumption also extends to the availability and semi-trustworthiness of the key screw providers.

Our proposed model defines an architecture in which no single entity, with the exception of the user's digital wallet has access to the entire cryptographic keys, and collaboration among less than t key escrow parties yields unsuccessful reconstruction of the secret keys. To improve security, the user's digital wallet supports the ability to store one or more key shares within the mobile device, acting as a key escrow provider. These shares can be protected using user's biometrics. As the minimum number of participants t increases, the

security of the system increases, however efficiency may be affected. The increase in number of participants P has an inverse affect on the efficiency of the system as well.

A key recovery mechanism should be deployed with great care. If the protocol is not correctly deployed it may lead to weaken security. The fundamental security requirement of our key recovery model should be that the effort to attack and exploit our model should be not less than the effort required to break a traditional key recovery protocol that relies on a single trusted third party [25].

Finally, in order for the user's digital wallet to encrypt each bundle, the wallet should have access to the public key of the receiving party. Our model assumes a decentralized trust model based on distributed ledger technology that facilitates the ability to look-up the public keys and service access points of key escrow providers. Our model further assumes that the wallet has the capability of verifying the identity of the key escrow providers through secure channels.

V. Implementation and The Next Steps

To implement the proposed model we rely on Hyperledger Indy [26]. Hyperledger Indy is an open source distributed ledger technology (DLT) developed specifically for self-sovereign identity use cases. The DLT is used as the root of trust where every entity of the network has a unique decentralized identifier. Decentralized Identifier (DID) is an emerging standard for generating unique identifiers that resolve to machine readable documents known as DID Documents (DDO) [27]. The DDO document consists of an entity's public keys and public service access points. The DDO document can also store the access points related to our secret sharing protocol. We develop our digital wallet on the Android platform using Java and Android Studio IDE. The digital wallet software consists of various components including a) agent, b) cryptographic libraries, c) identity store, and d) key store. Figure 2 depicts the high level architecture of our model. The agent component is responsible for establishing secure communication channels with other agents. A practical Indy architecture relies on cloud agents that provide a fixed public address to establish reliable communication. The cryptographic library includes the necessary cryptographic implementations such as Shamir's secret sharing algorithm. The Identity store and the keystore hold the identity information and the cryptographic keys respectively. The key escrow providers manage their own version of enterprise wallet with similar components.

As the next step we intend to fully develop the digital wallet software and create a proof-of-concept network of Hyperledger Indy nodes. We are also exploring the possibility of enhancing our secret sharing algorithms to improve the privacy and security of our model. Lastly, we intend to perform a security evaluation of our system architecture.

VI. Conclusion

Development of secure and practical key recovery solutions is paramount to the adoption of self-sovereign identity solutions. Driven by data confidentiality requirements and the increasing awareness around privacy, cryptographic enabled privacy schemes such as secure multi-party computation and threshold secret sharing techniques are gaining increasing popularity. This paper proposes a key recovery model based on Shamir's threshold secret sharing scheme and Hyperledger Indy. In this model a secret key is split into multiple shares and distributed among registered parties. The secret key is recovered successfully when a specified number of shares are retrieved by the digital wallet.

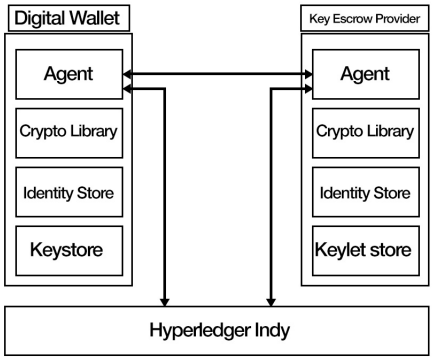

Fig. 2. Digital wallet architecture

References

[1] "1.1 billion 'invisible' people without ID are priority for new high level advisory council on identification for development". In: (Oct. 2017). URL: https://www.worldbank.org/en/news/press-release/2017/10/12/11-billion-invisible-people-without-id-are-priority-for-new-high-level-advisory-council-on-identification-for-development.

[2] "Blockchain and Financial Inclusion, The role blockchain technology can play in accelerating financial inclusion". In: Chamber of Digital Commerce, Mar. 2017.

[3] *LastPass Reveals 8 Truths about Passwords in the New Password Exposé. - The LastPass Blog*. URL: https://blog.lastpass.com/2017/11/lastpass-reveals-8-truths-about-passwords-in-the-new-password-expose.html/ (visited on 02/10/2019).

[4] F. G. Mármol, Joao Girao, and Gregorio M Pérez. "TRIMS, a privacy-aware trust and reputation model for identity management systems". In: *Computer Networks* 54.16 (2010), pp. 2899–2912. ISSN: 1389-1286.

[5] Christopher Allen. "The path to self-sovereign identity". In: *Life with Alacrity* (2016).

[6] *RWOT4 in Paris, France (April 2017). Contribute to WebOfTrustInfo/rwot4-paris development by creating an account on GitHub*. original-date: 2017-02-27T18:16:15Z. Apr. 26, 2019. URL: https://github.com/WebOfTrustInfo/rwot4-paris (visited on 04/29/2019).

[7] Jan-Erik Ekberg, Kari Kostiainen, and N Asokan. "Trusted execution environments on mobile devices". In: Nov. 2013, pp. 1497–1498. DOI: 10.1145/2508859.2516758.

[8] Túlio Cicero Salvaro de Souza, Jean Everson Martina, and Ricardo Felipe Custódio. "Audit and backup procedures for hardware security modules". In: *Proceedings of the 7th Symposium on Identity and Trust on the Internet*. ACM. 2008, pp. 89–97.

[9] Albert Bodo. "Method for producing a digital signature with aid of a biometric feature". In: *German patent DE* 42.43 (1994), p. 908.

[10] C Soutar and GJ Tomko. "Secure private key generation using a fingerprint". In: *Cardtech/Securetech Conference Proceedings*. Vol. 1. 1996, pp. 245–252.

[11] Benny Pinkas. "Cryptographic techniques for privacy-preserving data mining". In: *ACM Sigkdd Explorations Newsletter* 4.2 (2002), pp. 12–19.

[12] Andrew Chi-Chih Yao. "Protocols for secure computations". In: *FOCS*. Vol. 82. 1982, pp. 160–164.

[13] Oded Goldreich, Silvio Micali, and Avi Wigderson. "How to play any mental game". In: *Proceedings of the nineteenth annual ACM symposium on Theory of computing*. ACM. 1987, pp. 218–229.

[14] Benny Chor and Niv Gilboa. "Computationally private information retrieval". In: *Journal of the ACM*. Citeseer. 1997.

[15] Yehida Lindell. "Secure multiparty computation for privacy preserving data mining". In: *Encyclopedia of Data Warehousing and Mining*. IGI Global, 2005, pp. 1005–1009.

[16] D. G. Nair, V. P. Binu, and G. S. Kumar. "An effective private data storage and retrieval system using secret sharing scheme based on secure multi-party computation". In: *2014 International Conference on Data Science Engineering (ICDSE)*. 2014, pp. 210–214. DOI: 10.1109/ICDSE.2014.6974639.

[17] Adi Shamir. "How to share a secret". In: *Communications of the ACM* 22.11 (1979), pp. 612–613.

[18] Chris Clifton et al. "Tools for privacy preserving distributed data mining". In: *ACM Sigkdd Explorations Newsletter* 4.2 (2002), pp. 28–34.

[19] D Kumar Mishra, Rashid Sheikh, and Beerendra Kumar. "Privacy-preserving k-secure sum protocol". In: *IJCSIS International Journal of Computer Science and Information Security* 6 (2009).

[20] F. A. N. Pathak and S. B. S. Pandey. "An efficient method for privacy preserving data mining in secure multiparty computation". In: *2013 Nirma University International Conference on Engineering (NUiCONE)*. 2013, pp. 1–3. DOI: 10.1109/NUiCONE.2013.6780075.

[21] Alfredo De Santis et al. "How to share a function securely". In: *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*. ACM. 1994, pp. 522–533.

[22] George Robert Blakley. "Safeguarding cryptographic keys". In: *Proceedings of the national computer conference*. Vol. 48. 313. 1979.

[23] Zeng Yingli and Liu Dan. "A key escrow scheme to IOT based on Shamir". In: *2013 International Conference on Communications, Circuits and Systems (ICCCAS)*. Vol. 2. IEEE. 2013, pp. 94–97.

[24] Wenzhe Sun and Michiko Harayama. "A Proposal of Key Recovery Mechanism for Personal Decryptographic Keys". In: *2011 International Conference on Internet Technology and Applications*. IEEE. 2011, pp. 1–6.

[25] Konstantinos Rantos and Chris J Mitchell. "Matching key recovery mechanisms to business requirements". In: *Computers & Security* 24.3 (2005), pp. 232–245.

[26] *Hyperledger Indy - Hyperledger*. URL: https://www.hyperledger.org/projects/hyperledger-indy (visited on 03/07/2019).

[27] *Decentralized Identifiers (DIDs) v0.11*. URL: https://w3c-ccg.github.io/did-spec/ (visited on 03/02/2019).

View publication stats